International Journal of
Geo-Information

MDPI

Article

An Ontology-Driven Cyberinfrastructure for Intelligent Spatiotemporal Question Answering and Open Knowledge Discovery

Wenwen Li [1,*], Miaomiao Song [2] and Yuanyuan Tian [1]

[1] School of Geographical Sciences and Urban Planning, Arizona State University, Tempe, AZ 85287, USA; ytian72@asu.edu

[2] Institute of Oceanographic Instrumentation, Qilu University of Technology (Shandong Academy of Sciences), Qingdao 266061, China; mmsong@qlu.edu.cn

* Correspondence: Wenwen@asu.edu

Received: 10 October 2019; Accepted: 31 October 2019; Published: 3 November 2019

Abstract: The proliferation of geospatial data from diverse sources, such as Earth observation satellites, social media, and unmanned aerial vehicles (UAVs), has created a pressing demand for cross-platform data integration, interoperation, and intelligent data analysis. To address this big data challenge, this paper reports our research in developing a rule-based, semantic-enabled service chain model to support intelligent question answering for leveraging the abundant data and processing resources available online. Four key techniques were developed to achieve this goal: (1) A spatial and temporal reasoner resolves the spatial and temporal information in a given scientific question and enables place-name disambiguation based on support from a gazetteer; (2) a spatial operation ontology categorizes important spatial analysis operations, data types, and data themes, which will be used in automated chain generation; (3) a language-independent chaining rule defines the template for input, spatial operation, and output as well as rules for embedding multiple spatial operations for solving a complex problem; and (4) a recursive algorithm facilitates the generation of executive workflow metadata according to the chaining rules. We implement this service chain model in a cyberinfrastructure for online and reproducible spatial analysis and question answering. Moving the problem-solving environment from a desktop-based environment onto a geospatial cyberinfrastructure (GeoCI) offers better support to collaborative spatial decision-making and ensures science replicability. We expect this work to contribute significantly to the advancement of a reproducible spatial data science and to building the next-generation open knowledge network.

Keywords: spatial decision support; scientific workflow; provenance; reproducibility; open knowledge discovery; cyberinfrastructure; knowledge graphs

1. Introduction

To accelerate the knowledge discovery process and foster interdisciplinary research, it is important to move today's science paradigm toward becoming more open, collaborative, and reproducible. Open science, an effort to enable open access and open source to scientific data and research products, is an important endeavor to ensure transparency in the creation and dissemination of scientific knowledge [1]. In geospatial domains, much progress has been made. First, many government agencies and organizations are increasingly sharing the geospatial data products acquired from earth observation platforms, sensor networks, and field surveys. These data constitute important resources for deriving new knowledge and insight about the physical world and society [2]. Second, open source geospatial analysis libraries, such as the Geospatial Data Abstraction Library (GDAL) and Python Spatial Analysis Library (PySAL), have been developed to power-up existing systems with spatial analytics capabilities. Third, the rapid advancement of geospatial cyberinfrastructure [3], which seeks to provide better organization, integration, computation, and visualization of georeferenced resources, thereby enabling virtual collaborative research on an unprecedented scale.

Despite these advances, significant challenges remain to the interoperation, coordination, and integration of distributed resources to enable "smart" knowledge discovery. The vast amounts of available data that have been collected or created have become increasingly heterogeneous—they differ in data organization, encoding, spatial reference system, and level of detail in metadata, as do the spatial analysis resources [4]. They differ in naming conventions, input, and output structures, and programming language as well as implementation of algorithms. This heterogeneity has created a very steep and long learning curve for non-experts to conduct spatial analysis and resolve domain-specific questions. To address this issue, a service-oriented paradigm has been proposed [5], in which all the geospatial resources are encapsulated as standard web services. By defining a unified protocol for interacting with a remote web service hosting data or geoprocessing functions, a loosely coupled problem-solving environment can be established [6].

In addition, geospatial ontologies, or knowledge graphs, are created to add smartness to a system to aid the machine understanding of data content and contexts of a geoprocessing function [7]. Ontologies are formal modellings of knowledge; they define a hierarchical classification of concepts and interrelations among them. Knowledge graph can be considered an instantiation of domain knowledge leveraging the schema defined in an ontology. Semantic inference can then be performed on top of the knowledge graphs to find concepts and hidden linkages of interest. Aid by an ontology, a system can identify the most suitable data that meet the spatial, temporal, and semantic needs for answering a scientific question. However, many queries performed on top of an ontology remain within the scope of knowledge encoded in it [8]. The ability of using an incomprehensive ontology to infer new knowledge, especially new spatial knowledge, is limited [9]. It therefore becomes utterly important to develop a new mechanism to integrate spatial analytics and link a dataset to a spatial analytical function and/or link between spatial analytical functions for the dynamic generation of scientific workflows towards intelligent analysis and question answering [10].

Another challenge for current cyberinfrastructure portals, or the majority of available Geographic Information Systems (GIS), is the very weak capability for tracking the flow of data—what data is used, which spatial analytical function it is fed in, how the output result is presented, and so on [11]. This limitation has significantly hindered the ability to reproduce and validate one's own or others' work [12]. The consequence of this lack of reproducibility is massive amounts of duplication in the efforts for data preparation, method, and workflow development. Hence, automated recording of the workflow metadata and relevant provenance information in existing systems will become an important milestone for open and reproducible science.

This paper introduces our work in tackling the aforementioned challenges by developing a cyberinfrastructure to support intelligent question answering and open knowledge discovery. Advanced capabilities are developed to enable a knowledge creation process in which the queries can be interpreted and relevant data can be collected, chained, analyzed, and visualized to support spatial decision-making. The remainder of the paper is organized as follows: Section 2 provides a review on geospatial interoperability, ontology-based service chaining, and workflow generation. Section 3 describes a scientific use case on earthquake data query. Section 4 introduces our proposed method. Section 5 discusses the implementation details of the system. Section 6 concludes the paper and discusses future research directions.

2. Literature Review

2.1. Data and Geoprocessing Services

Geoprocessing and spatial data services are popular means for sharing and analyzing geospatial data in a distributed computing environment [13,14]. A web service helps general users get access to ready-to-use information to obtain geospatial data and paves the path from data to useful information and knowledge [15]. The services of the Open Geospatial Consortium (OGC), which provides both a high-level abstraction and detailed implementation specifications for sharing and requesting remote data and processes, are widely adopted by different organizations to enhance geospatial interoperability [16]. OGC provides a variety of web service standards, such as Web Feature Service (WFS) for sharing vector data, Web Map Service (WMS) and Web Map Tile Service (WMTS) for sharing data rendered as (tiled) maps, Web Coverage Service (WCS) for sharing gridded data, Sensor Observation Service (SOS) for sharing sensor observation data, and Web Processing Service (WPS) for sharing geospatial processes to enable the reuse of analytical functions. OGC services have found widespread use in building national and international spatial data infrastructure [16–20], as well as supporting disaster management [21,22], monitoring hydrological conditions [23], and conducting urban planning [24] and policy-making [25].

2.2. Ontology Support for Building a Geoprocessing Framework

Instead of using a single geoprocessing service, tackling a complex, real-world problem often requires the integration and chaining of multiple geoprocessing and data services [13,26]. Service chaining links multiple services together to generate a feasible workflow. It differs from service composition in that the latter has a flatter and more static structure, with the order and position of a service in a chain pre-known, so all the services are composed together all at once. Instead, service chaining consists of linking the data and processing service in a more dynamic and on-demand manner. This requires the chaining engine to be able to understand the input and output of a geoprocessing service at both the syntactic and semantic levels [27]. Ontology, a formally defined and machine-understandable knowledge base, is often used to define data type, service type, and associations among geoprocessing services to feed the correct data into a geoprocess and aid the chaining of multiple geoprocesses [28–30]. GeoBrain (http://www3.csiss.gmu.edu/OnAS/) develops an ontology model and uses semantic referencing and matching to generate an executable workflow for online geospatial analysis [31]. Li et al. [32] use the domain ontology Semantic Web for Earth and Environmental Terminology (SWEET) to guide service metadata parsing and service content comprehension to support the composition of highly diversified and distributed data and geoprocessing resources for building an Arctic Spatial Data Infrastructure (Arctic SDI). Al-Areqi et al. [33] proposed a semi-automatic semantics-based workflow designed to evaluate sea level rise. Despite these efforts in providing semantic annotations on data and services, an integration ontology that generalizes workflows and integration rules remains highly demanding to enable more flexible and smarter service understanding and chaining [34,35].

2.3. Service-Oriented Workflow Technologies

Several workflow languages and technologies have been developed to support geoprocess composition and chaining. Here we focus on reviewing those with the capabilities to support *service-oriented* computing because such technology enables seamless integration of data and processing resources from physically distributed locations, ensuring the reuse of valuable resources and scalability of an analytical system.

Enabling automated web-service-based workflow generation and execution has two required elements: (1) metadata or workflow language that can track the provenance of a chain of geoprocessing and (2) a workflow engine that can parse the metadata file and execute the workflow step by step by binding the right data to the right process. Business Process Execution Language (BPEL) is a workflow language initially used in business [36] and later introduced to support geoprocessing orchestration. Many earlier works in chaining OGC services have utilized BPEL. For instance, Brauner et al. [37] wrap GIS functions from GRASS into a WPS and use BPEL to encode service chain. Yu et al. [38] develop a BPEL execution engine that provides powerful geospatial support in terms of handling Geography Markup Language (GML), WFS, and WPS. However, BPEL shows limitations in supporting binary data types [21]; in addition, it requires the creation of Web Services Description Language (WSDL) documents for each service in the chain, complicating system implementation. Zhang et al. [39] developed a standalone workflow tool that provides an interactive interface to allow web service orchestration by end users. This tool provides flexible workflow customization like ArcGIS's module builder. Although great progress has been made, current workflow generation still involves manual work or is based on a hard-coded workflow.

It is the aim of this paper to tackle the aforementioned problems by developing (1) an ontology-based, service-oriented method that enables smart and automated matching between data and processes and chaining between processes; (2) a rule-based chaining algorithm to allow dynamic and on-demand chaining and execution of web services to ensure elastic and adaptive problem solving; and (3) a cyberinfrastructure that integrates the workflow generation and execution module as well as data mapping and visual analytics to enable a replicable research paradigm.

3. A Natural Disaster Use Case

Figure 1 demonstrates an earthquake use case where a researcher or a local resident is interested in retrieving information about all earthquakes that occurred in California from January to March last year. In a geographical-information-centered platform, it would be most useful to provide not only the number of earthquakes but also their locations, time of occurrence, and magnitude, as well as a series of analysis functions to offer the flexibility to generate statistics based on the interest dimensions of information. In addition, dynamic mapping will be a desired feature for presenting the earthquake information vividly on a map for easy information comprehension. It would also be helpful for the platform to capture the analytical workflow made by an end user and allow the same or a different user to easily reproduce the results using the saved workflow on the platform.

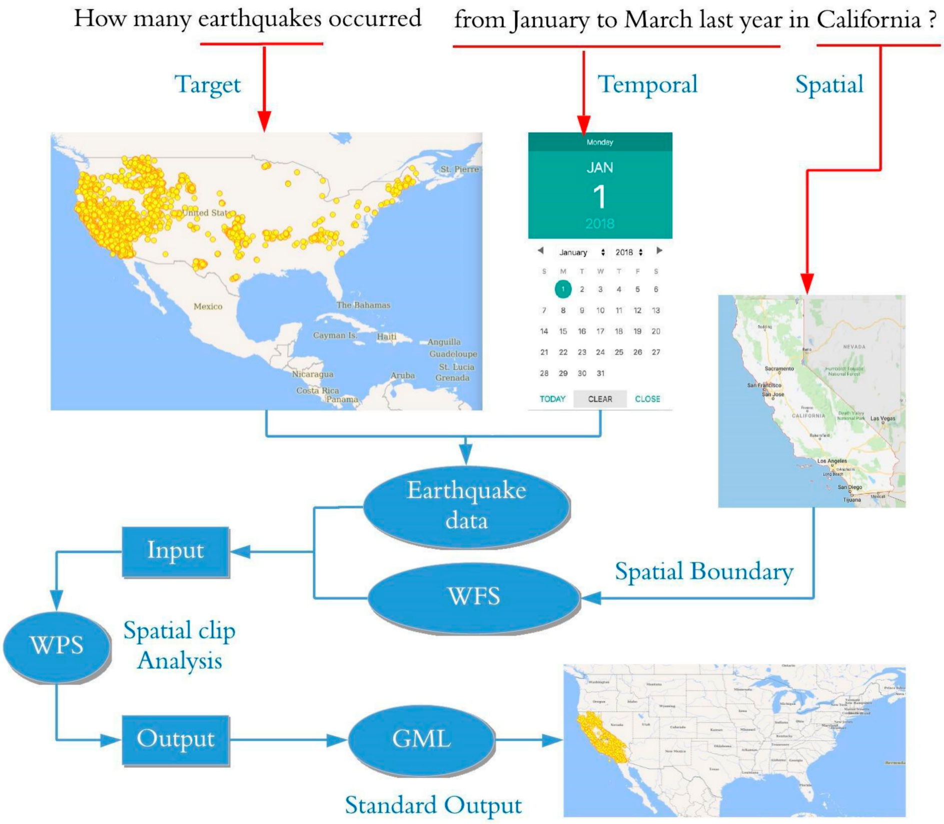


Figure 1. A natural disaster use case—an inquiry about the occurrence of earthquakes in California.

Technically, once the question is posed, the intelligent agent should be able to parse important components (space, time, and theme) in the question and automatically search for data that fit the theme of interest, perform necessary processing on the data, such as spatial and temporal filtering, and create a workflow and execute it to deliver the results through on-demand spatial analysis. It is crucial for a system to have the ability to integrate and interoperate data from diverse sources and to link data with the necessary operations, which are also interoperable, to enable an automatic chain of analyses.

However, to date, there are very few systems that have such capabilities to meet the above requirements. In this paper, we will introduce our cyberinfrastructure solutions for this problem.

4. Semantic-Enabled Service Chain Model

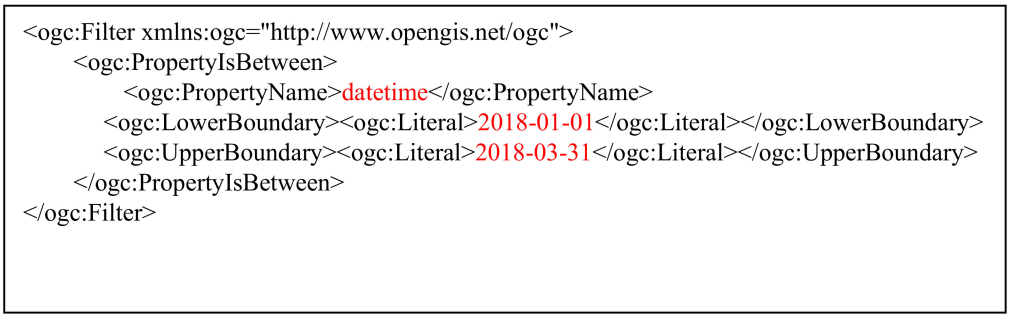
This section describes key components of the automated generation of an executable workflow for addressing complex spatiotemporal questions.

4.1. Spatial and Temporal Reasoning

Spatial and temporal reasoning are the keys to answering geospatial questions related to space and time. To accomplish this goal, a space-time reasoner is designed to transform place names and temporal keywords within a scientific question into machine-understandable spatial-temporal constraints. For instance, the spatial constraints could be the coordinates of a location described by its latitude and longitude or the boundary information of a place, interpreted from its name. The space-time reasoner comprises two components: spatial reasoner and temporal reasoner. In the spatial reasoner, a DBpedia knowledge base (KB) is adopted to mine the municipality level (i.e., country, state, city, or village) of a place name and perform place name disambiguation. Spatial reasoning is realized based on a gazetteer look-up tool, which is developed by DBpedia as a web service interface to query the DBpedia KB. Two parameters—“QueryClass” and “QueryString”—are required to construct the query, where “QueryClass” refers to an ontology class name (e.g., Person, Place, Species, etc.) and “QueryString” indicates a specific keyword, which is usually a place name for spatial reasoning. The municipality level of a place name is included in the “Categories” element of the query results in the format of Extensible Markup Language (XML). By parsing the query result and detecting elements that point to country or state or other municipality levels, the location information can be easily extracted. In the implementation of the spatial reasoner, the municipality level is used to determine which spatial (boundary) data should be implanted into a specific geospatial analysis. In this step, using place names as the query condition, accurate spatial boundary information will be retrieved from the pre-prepared boundary data in vector formats.

Temporal reasoning is accomplished by detecting the phrase regarding temporal elements from a spatial question, such as “from January to March last year”, and converting this temporal information to a standard time range. The Stanford Time Tagger SUTime, an open-source library implemented by Java code to recognize and normalize time expressions from text data in natural language, is adopted to perform temporal parsing [40]. By importing the SUTime package, temporal expressions can be extracted as temporal objects from the spatial question according to text rules defined as a regular expression. Then, temporal objects are normalized by a transformation to the format of yyyy-mm-dd. Finally, time range is represented as a query filter according to the OGC Filter specification by combining the start date with the end date using the operator of *PropertyIsBetween* (Figure 2).

In this way, a query filter is generated and embedded into a service chain for determining which entities in spatial data with time series should be injected into a spatial analysis.



```
<ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
      <ogc:PropertyIsBetween>
            <ogc:PropertyName>datetime</ogc:PropertyName>
         <ogc:LowerBoundary><ogc:Literal>2018-01-01</ogc:Literal></ogc:LowerBoundary>
         <ogc:UpperBoundary><ogc:Literal>2018-03-31</ogc:Literal></ogc:UpperBoundary>
      </ogc:PropertyIsBetween>
</ogc:Filter>
```

Figure 2. The result of time reasoning encoded according to the OGC Filter Specification.

4.2. An Ontology of Spatial Analytical Methods

In this part, an ontology is established for linking spatial operations and data. This ontology provides a conceptual model for building analysis rules to conduct a service chain automatically. The ontology of spatial analytical methods consists of three components, spatial operation ontology, spatial data ontology, and theme ontology, as shown in Figure 3.

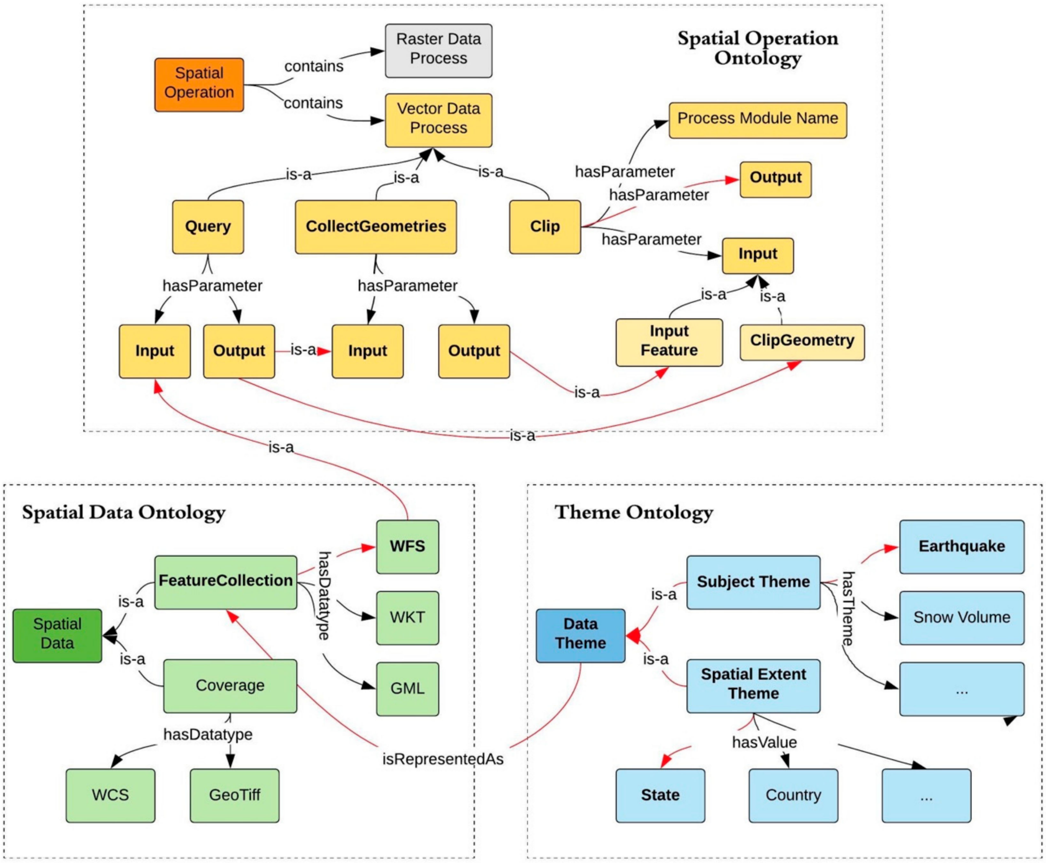


Figure 3. The ontology of spatial analysis methods. All the ontological elements used for clip analysis for the Earthquake use case are highlighted in bold text. The relationships between these ontological elements are highlighted in red. Theme ontology encodes the subject theme "Earthquake" and the spatial extent theme "State," and both data will be represented as a FeatureCollection (in Spatial Data Ontology) and made available as a WFS. This WFS will serve as the input for temporal and spatial queries defined in the Spatial Operation Ontology. The CollectGeometries will combine discrete spatial extent geometries into a single geometric object used as the input for Clip analysis. The Earthquake WFS will be fed into Clip operation as Input Feature. The output from this analysis will be desired query results.

Spatial operation ontology is constructed for the semantic understanding of geo-processes, and it contains two classes: raster data processes and vector data processes. Every process has the properties of a module name, input, and output. A module name is the unique identifier of a process extracted from a service capability file from a remote OGC WPS. When different services perform the same spatial operation, they are differentiated by their service location plus process name. The input and output of a module are instances of a spatial data ontology; they could be either numerical or literal values, or a spatial data with various types. The output of one process may be the input of others.

In a spatial data ontology, data are classified into two types: coverage for representing raster data and feature collections for representing vector data. Each type possesses the properties of "data type" and "theme name". Data types of coverage include WCS and GeoTiff File, while data types of feature collections contain Well-known Text (WKT), Geography Markup Language (GML), Geo-JavaScript Object Notation (GeoJSON), and WFS. A theme name is used to link spatial data ontology with theme ontology.

Theme ontology defines the semantics of keywords related to the query subjects in spatial questions. Partial themes are listed in Figure 3, including earthquake and snow volume, which integrate several geoscience use cases. All the themes have a root node: Subject Theme. and location information, such as the boundary of administrative regions and places, described in Spatial Extent Theme.

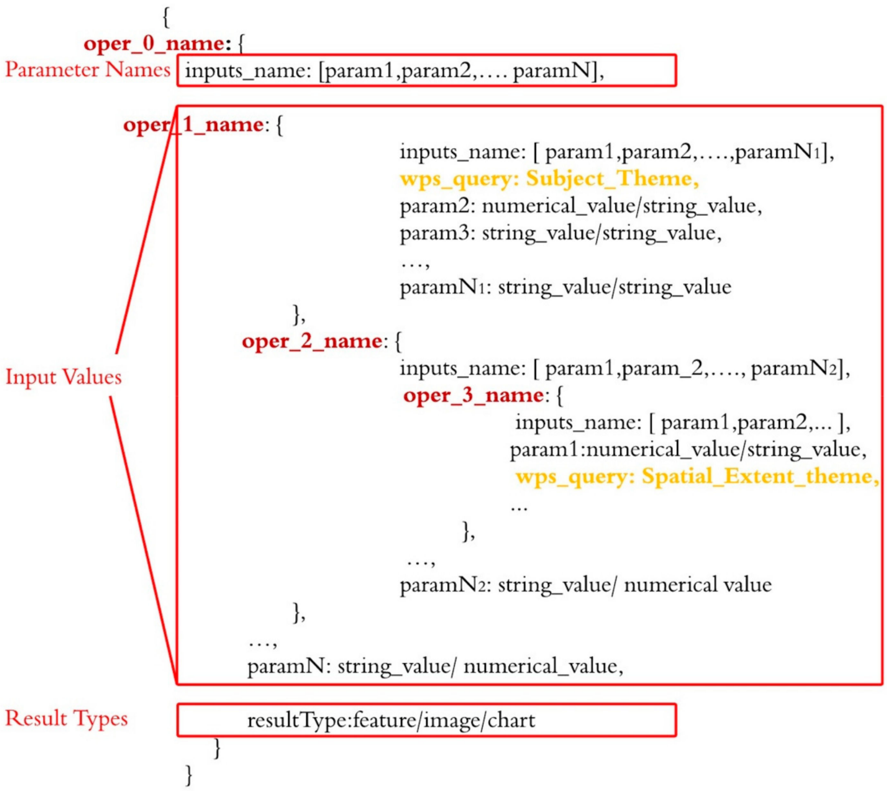
In Figure 3, we also provide details of the clip analysis to demonstrate how spatial analysis ontology, spatial data ontology, and theme ontology act together to address a spatial question. Clip belongs to a type of vector analysis and it has two inputs, Input Feature and ClipGeometry. Input Feature contains a collection of features as the clip target and ClipGeometry refers to a geometry containing several spatial areas (e.g., a multi-polygon) to use for clipping. Spatial data required by Input Feature and ClipGeometry are defined in the spatial data ontology. Use cases that can be benefitted from this analysis are defined in the theme ontology. The input data of a spatial operation, however, may not be directly available; instead, it may need to be derived from another dataset and analysis before it can be used for this analysis. For the Clip operation, the value of Input Feature is the output of the Query operation, which is used to query spatial data through a WFS with a Subject Theme (e.g., earthquake) and returns a set of eligible features (e.g., earthquake points). Meanwhile, the value of ClipGeometry is the output of the CollectGeometries operation, which converts a set of features into a composited geometry using the output of the Query operation as input. Eventually, for moving data into ClipGeometry, the Query operation will act on top of a WFS, which hosts spatial boundaries to get the interested geographical region using a filter defined in the Spatial_Extent_Theme (e.g., state or country). In Figure 3, the ontological components and their linkages needed for the earthquake use case are highlighted with bold text and red arrow respectively.

In the semantic-enabled service chain model, spatial analysis methods and data are organized according to the above ontological structure. For implementation, these data and ontological resources are stored in the PostgreSQL database. Implementation details are elaborated in Section 5.

4.3. Defining Rules for Automated Service Chaining

The chaining rule is defined to enable automated service chaining and workflow generation. In our solution, these rules are defined in a language-independent, machine-understandable format of JavaScript Object Notation (JSON), such that it can be easily parsed or generated using different programming and workflow languages.

Figure 4 illustrates a universal schema of the rule to enable complex spatiotemporal analysis in a nested structure. The item located in the outermost layer is considered to be the top-level operation and represented as a key-value pair, where the key is "*oper_0_name*", which is the name of an operation provided by a WPS or other type of analytical services. All of the JSON objects following the colon act as the value for key "*oper_0_name*" and include definitions for input names, input values, and the results type.



```
{
  oper_0_name: {
    inputs_name: [param1,param2,…. paramN],
      oper_1_name: {
                      inputs_name: [ param1,param2,….,paramN1],
                      wps_query: Subject_Theme,
                      param2: numerical_value/string_value,
                      param3: string_value/string_value,
                      …,
                      paramN1: string_value/string_value
             },
         oper_2_name: {
                      inputs_name: [ param1,param_2,…., paramN2],
                      oper_3_name: {
                              inputs_name: [ param1,param2,... ],
                              param1:numerical_value/string_value,
                              wps_query: Spatial_Extent_theme,
                              ...
                        },
                       …,
                      paramN2: string_value/ numerical value
             },
         …,
         paramN: string_value/ numerical_value,
         resultType:feature/image/chart
    }
}
```

Figure 4. The universal representation of the rule for complex spatiotemporal analysis.

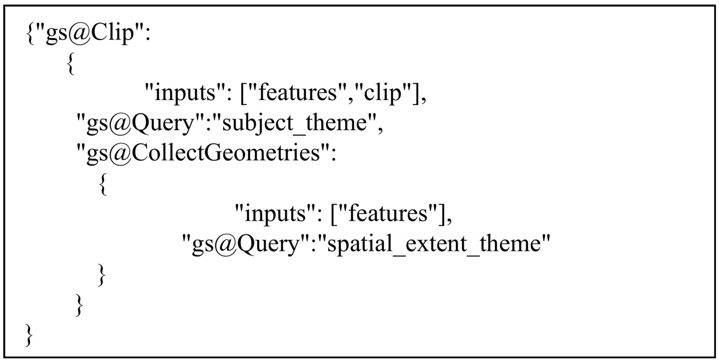
The component of *input_names* lists the names of all parameters in a certain order. Following the parameters list, there are several key-value pairs acting as input values, the order of which is the same as the index of parameters. The last element of the top-level operation is *resultType*, which indicates the ways that the results will be represented. They could be images, feature sets, or statistics of the raw results. The input values could either be some existing dataset, available as part of a web service, or the derived products from another geoprocessing operation. The former is a type of direct input, and the latter is a derived input. The direct input can be provided as a string, indicating the layer name of data from a web service, or it could just be a numerical value used as the parameter of some analytical functions. The derived input will invoke another process, which has the same structure of the top-level process illustrated in Figure 3. For instance, when the input data is only a spatial or temporal subset of the original data, a corresponding filter (which is another process) will be applied to derive the needed dataset.

The *oper_1_name* and *oper_2_name* gives examples for defining a built-in process within the parent process *oper_0_name*, and *oper_3_name* is another built-in function defined within *oper_2_name*, generating a nested structure for tackling a complex spatiotemporal question.

In Figure 4, we define "*wps_query*" (marked in yellow) as a special function used to inject spatial data into a service chain. "*wps_query*" is always put in the deepest level of an *input_value* branch with a theme name as its value. The possible theme names are defined in the theme ontology and they fall into two types: *Subject_Theme* and *Spatial_Extent_Theme*. When the value of the "*wps_query*" is a subject theme, relevant thematic data with time series, such as earthquake data, will be bound to this operation. When the value is about a spatial extent, corresponding spatial extent data (such as country or state boundaries) will be bound to the operation. In Section 5, we will elaborate how the data can be loaded dynamically as the service chain is being generated.

Based on the above description, we formalize the principle used to construct an analysis rule as follows: (1) every spatial operation can be treated as a top-level operation or a built-in operation; (2) a top-level operation can take the output of a built-in operation as its input; (3) a built-in operation can take the output of another built-in operation as input; and (4) any branch should end up with the operation "*wps_query*". The spatial operations do not need to be implemented and hosted locally. Instead, they are expected to be made available in a standard web service format, such as OGC WPS, to enable reuse and interoperability among remote service providers. Following this principle, it is not only possible to make the analysis rule extensible and scalable but also capable of automatically generating an executable workflow through a nested service chain.

Figure 5 demonstrates the application of the analysis rule to define the clip operation named "gs@Clip".



```
{"gs@Clip":
     {
              "inputs": ["features","clip"],
      "gs@Query":"subject_theme",
      "gs@CollectGeometries":
         {
                         "inputs": ["features"],
                  "gs@Query":"spatial_extent_theme"
         }
      }
}
```

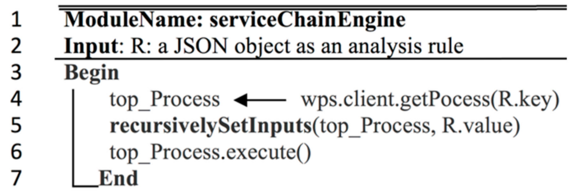
Figure 5. An example of rule definition for the clip operation in JSON format.

Two input values of the operation "gs@Clip" are, respectively, the outputs of operations for "gs@Query" and gs@CollectGeometries; the gs@CollectGeometries operation embeds another "gs@Query". Finally, specific spatial data, defined in subject_theme and spatial_extent_theme, are bound to operations named "gs@Query", which are located at the leaf nodes. In this way, data resources encoded in the spatial data ontology can be fed into the corresponding processing services defined in the spatial operation ontology to create a geo-service chain following the analysis rule. In practice, the theme element and spatial and temporal elements of a spatial question will be identified to select and apply the related data. For instance, in the spatiotemporal question given in the use case "how many earthquakes occurred in California from January to March last year?", "earthquake" will be identified as the theme element, "California" will be identified as the spatial element, and "January to March last year" is a time element and will be transformed to a date range by a time reasoner for further processing.

4.4. A Recursive Algorithm for Generating Executable Workflow

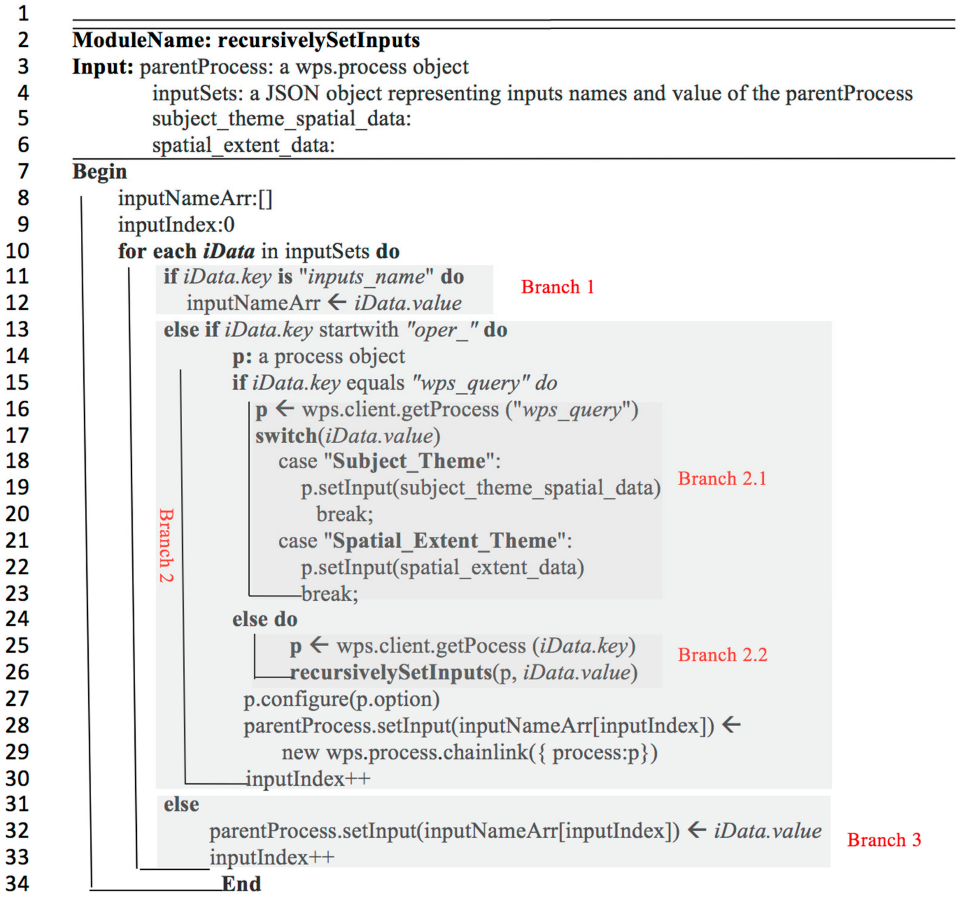
To parse an analysis rule and generate an executable workflow, a recursive algorithm is designed to build a service chain compatible with OGC WPS, because it is a web service standard supported widely by many web servers.

The algorithm consists of modules of *serviceChainEngine* (Figure 6) and *recursivelySetInputs* (Figure 7). In *serviceChainEngine*, an object of *wps.process* is created using *wps.client.getProcess()* with the processing name at the top-level operation as the key. Next, the module of *recursivelySetInputs* is invoked to substantiate inputs recursively for the top-level process to generate a complete and executable workflow metadata file. The metadata will then be sent to a WPS engine for generating the final results.



```
ModuleName: serviceChainEngine
Input: R: a JSON object as an analysis rule
Begin
    top_Process ←— wps.client.getPocess(R.key)
    recursivelySetInputs(top_Process, R.value)
    top_Process.execute()
End
```

Figure 6. Algorithm for generating and executing a service chain.



```
ModuleName: recursivelySetInputs
Input: parentProcess: a wps.process object
       inputSets: a JSON object representing inputs names and value of the parentProcess
       subject_theme_spatial_data:
       spatial_extent_data:
Begin
    inputNameArr:[]
    inputIndex:0
    for each iData in inputSets do
        if iData.key is "inputs_name" do                                   Branch 1
            inputNameArr ← iData.value
        else if iData.key startwith "oper_" do
            p: a process object
            if iData.key equals "wps_query" do
                p ← wps.client.getProcess ("wps_query")
                switch(iData.value)
                    case "Subject_Theme":
                        p.setInput(subject_theme_spatial_data)              Branch 2.1
                        break;
                    case "Spatial_Extent_Theme":
                        p.setInput(spatial_extent_data)
                        break;
            else do                                                         Branch 2
                p ← wps.client.getPocess (iData.key)                        Branch 2.2
                recursivelySetInputs(p, iData.value)
            p.configure(p.option)
            parentProcess.setInput(inputNameArr[inputIndex]) ←
                new wps.process.chainlink({ process:p})
            inputIndex++
        else
            parentProcess.setInput(inputNameArr[inputIndex]) ← iData.value  Branch 3
            inputIndex++
End
```

Figure 7. Algorithm for recursively setting the input of a process.

The module of *recursivelySetInputs* is the core to build a WPS service chain. It uses a *wps.process* object and a set of objects in JSON format as its inputs. By traversing the *inputSets* recursively, the key and value of each specific parameters will be parsed and handled. As shown in Figure 7, three cases are considered in parsing an analysis rule defined for a science use case.

- Branch 1: When the input of *iData.key* is an available dataset, its value will be stored in a local variable *inputNameArr* storing all the process execution metadata.
- Branch 2: If the key starts with "*oper_*", it means *iData* is not a direct input; instead, its value will be derived from another built-in function. In this case, a new process object *p* will be created. When the *iData.key* equals to "wps_query", a WPS query process will be created and spatial data indicating the subject or spatial extent will be bound as its input (Branch 2.1). When *iData.key* is not a "wps_query", it will be treated as a generic built-in operation. Any spatial analytical operation defined in the spatial operation ontology can be a candidate for this built-in function. To deal with this kind of complex analysis, the first step is to create a new process *p* and set *iData.key* as its process name. The second step is to call the *recursivelySetInputs* function itself to recursively feed the data into it. After all inputs for *p* are set properly, an object wps.process.chainlink from this process will be set as the input to its *parentProcess* saved in the *inputNameArr* (Branch 2.2).

- Branch 3: When *iData.key* is neither a directly available dataset (Branch 1) or a derived dataset from another operation (Branch 2), its value (*iData.value*) will be treated as a simple data type (a string or a numeric value) and set to its parent process listed in the corresponding order of the list *inputNameArr*. The local variable *inputIndex* saving the index of the input parameters will increase by one after the processing steps described in Branch 2 and 3.

It needs to be noted that although we use the generation of workflow metadata that is compliant with WPS as an example, the logic and algorithm here are general and can be applied to any other data or service standards that support embedded spatial analyses in a workflow.

5. Implementation

This section introduces the implementation of the service chain engine that can create, execute, and visualize the results for tackling a complex spatiotemporal question.

5.1. Architecture

The architecture for implementing automated service chaining with the support of spatiotemporal semantics is shown in Figure 8. Its components include an ontology database, a service chaining engine, an OGC Web Service engine and an interactive user interface (UI) for visual analytics. The ontology database stores meta-information of spatial data and processes, which are classified into rules, themes, spatial analysis methods, and a spatial data catalog according to the ontological structure defined in Section 4.3. The service chaining engine is the core component and is responsible for assembling spatial data and processes based on the spatial analysis rules. Built upon the algorithm described in Section 4.4, it parses and transfers the rules into an executable WPS instance by generating a normative WPS *Execute* request. The open source OGC web service engine, GeoServer, is adopted to host entities of spatial data and processes, which are registered into the ontology in advance. In addition, the service engine is also responsible for the execution of service chains. Finally, the analytical results from a complex workflow are visualized as maps (e.g., point, polygon, or polyline) and/or statistical chart (e.g., bar chart or line chart) in an interactive UI portal established in a geospatial cyberinfrastructure environment.

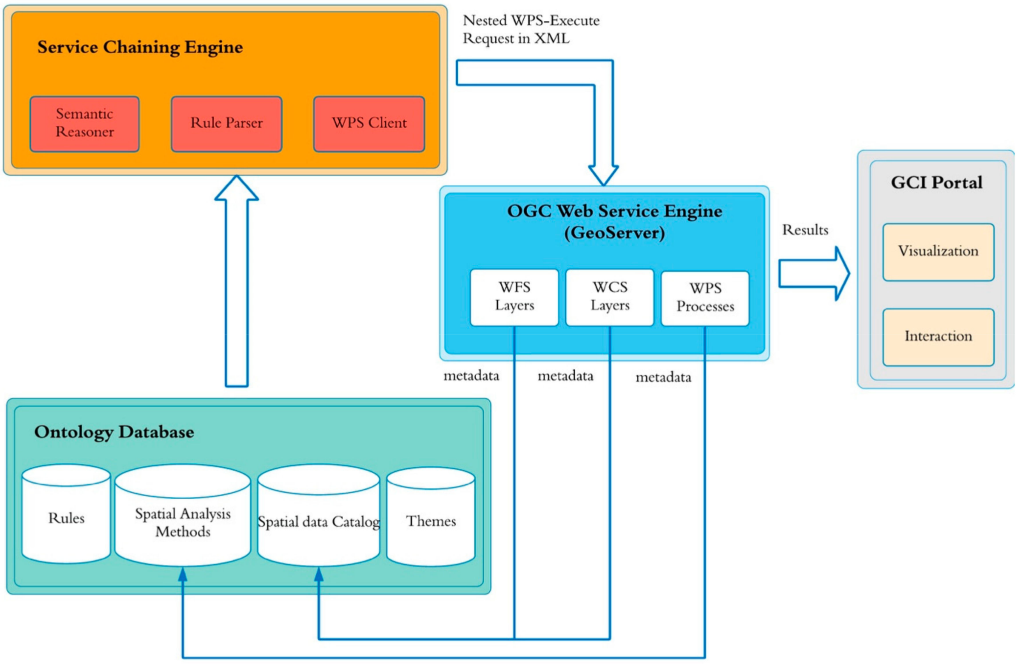


Figure 8. Architecture for the rule-based, semantic-enabled workflow generation, execution and visualization engine.

5.2. Ontology Database

As a core component, the ontology database is built to store metadata information and semantic annotation of themes, rules, and spatial data. Figure 9 demonstrates the logical view of the ontology database and the relationship among spatial data, rules, and themes. The "themes" table is designed to store semantic topics involved in spatial questions. It has the structure of *{theme_id, theme_name, analysis_name}*, where *theme_name* indicates the subject of the spatial data. The "rules" table, which defines the principles for building service chains, contains the attributes of *{analysis_name, analysis_rule}*. The spatial data catalog has a structure of *{theme_id, data_type, data_content, SRS, temporal_field, spatial_field}* with the combination of *theme_id* and *srs* as the primary key. Here, *data_type* refers to the type of spatial data (e.g., vector or raster), *data_content* stands for the layer name of the input spatial data, *srs* refers to spatial reference system, *temporal_field* indicates the property name of the spatial data used for temporal filtering, and *spatial_field* refers to the property name used for spatial filtering. The table of spatial data catalog is connected with the themes table by *theme_id* as the foreign key, while themes and rules are matched through the field of *analysis_name*.

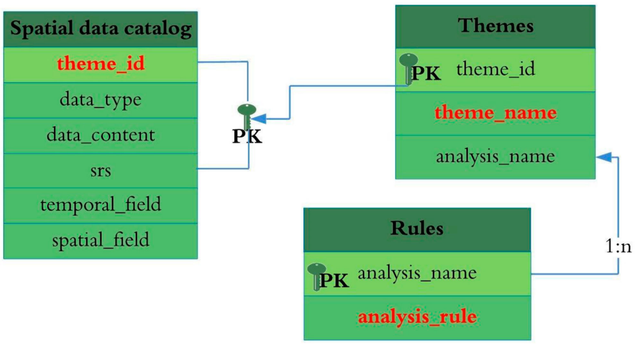


Figure 9. ER (Entity-Relation) model of the ontology database.

The ontology database provides principal meta-information to support the seamless linkage of related data and process to resolve specific spatial questions. The most important principle in establishing an ontology database is to ensure that spatial data and processes are semantically annotated, so that they can be more easily discovered and identified as related to solving a certain spatial problem.

5.3. Automated Service Chaining Engine

The intelligent analysis of geospatial data is mainly accomplished by the service chaining engine, through the steps of automatically conducting (1) semantic reasoning, (2) analysis rule parsing, (3) atomic services and spatial data assembling, and (4) executable service chains generating. Figure 10 illustrates the workflow of an automated services chaining engine.

By detecting the key elements <space, time, theme, analysis> from a spatial question, the service chaining engine triggers the time reasoner, rule parser, and spatial reasoner to conduct semantic reasoning in terms of space and time. First, the time reasoner builds the temporal filter using the start and end time given in the query. Next, the time filter will be applied to the data related to the theme of interest, such as earthquake, to obtain the subset of data falling within the given time period. An example of a temporal filter is shown in Figure 3. On the other hand, the spatial reasoner will perform place name disambiguation with the aid of the DBpedia gazetteer and then search the corresponding boundary file at the user-designated level, such as state or country, to obtain the boundary data. Figure 11 gives an example of the spatial filter compliant with the OGC Filter standard. Then some spatial analysis (in the earthquake case it is clip analysis) is applied to the temporally filtered thematic data (the earthquake data) to get a spatial subset of data within the region of interest.

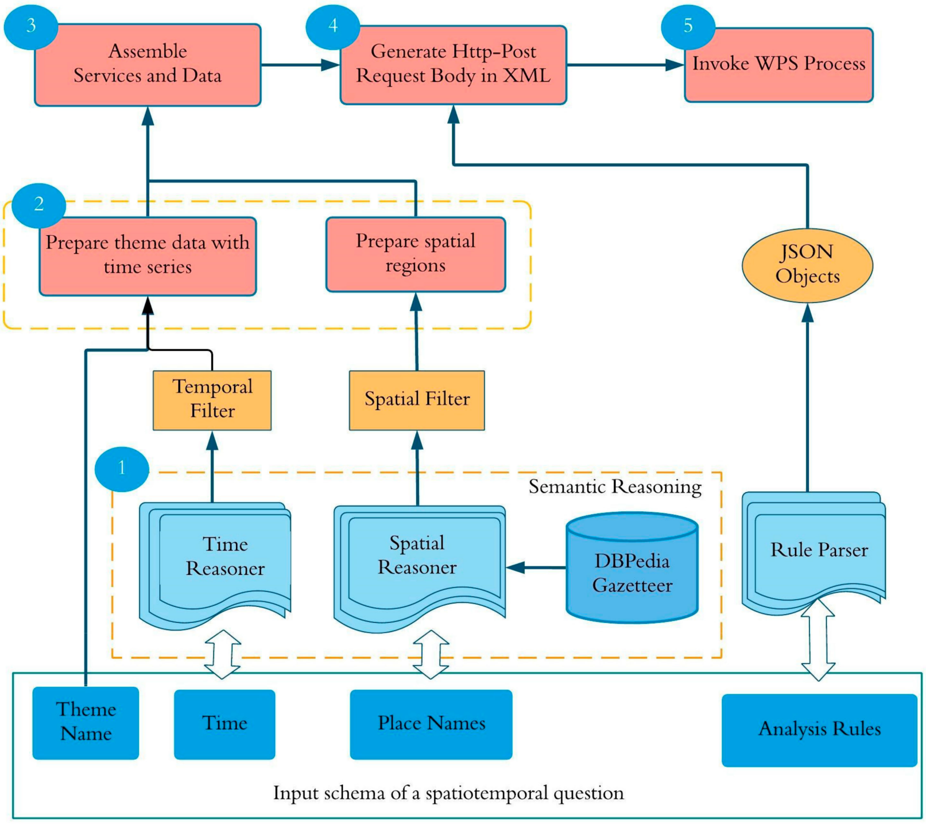
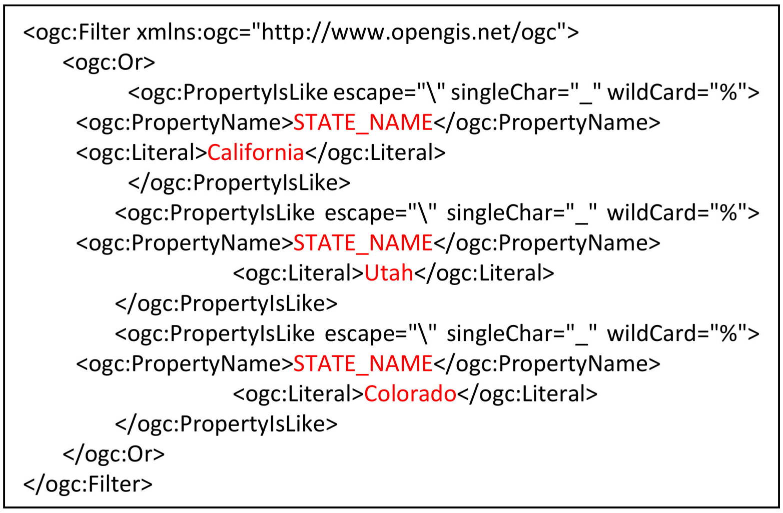


Figure 10. The workflow of the automated service chaining engine and question answering.



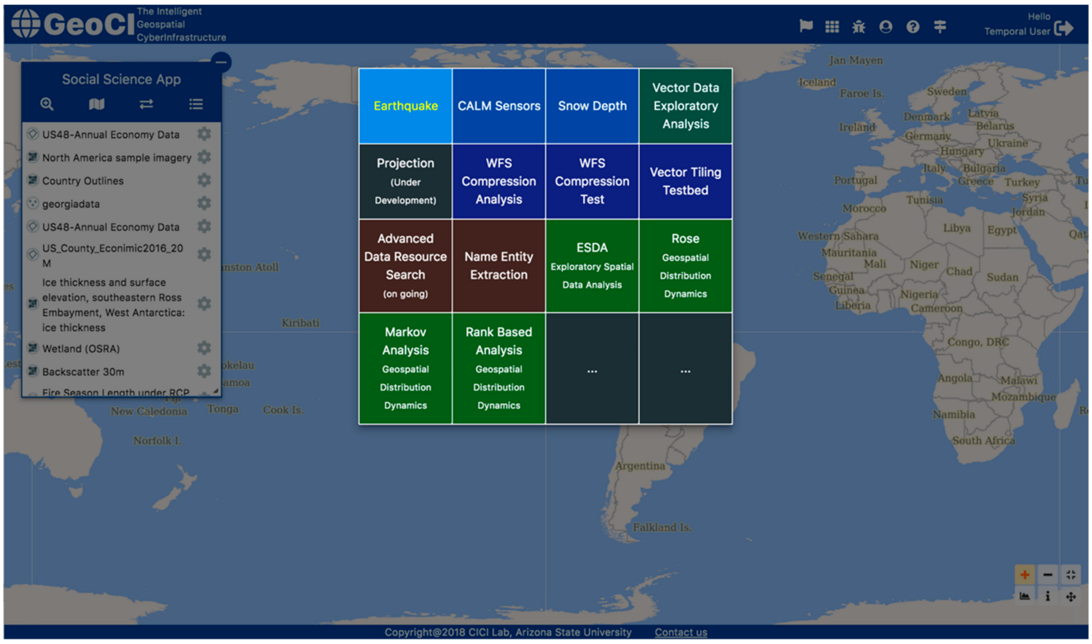
```
<ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
   <ogc:Or>
        <ogc:PropertyIsLike escape="\" singleChar="_" wildCard="%">
    <ogc:PropertyName>STATE_NAME</ogc:PropertyName>
    <ogc:Literal>California</ogc:Literal>
        </ogc:PropertyIsLike>
       <ogc:PropertyIsLike  escape="\"  singleChar="_"  wildCard="%">
    <ogc:PropertyName>STATE_NAME</ogc:PropertyName>
                <ogc:Literal>Utah</ogc:Literal>
       </ogc:PropertyIsLike>
       <ogc:PropertyIsLike  escape="\"  singleChar="_"  wildCard="%">
    <ogc:PropertyName>STATE_NAME</ogc:PropertyName>
                <ogc:Literal>Colorado</ogc:Literal>
       </ogc:PropertyIsLike>
   </ogc:Or>
</ogc:Filter>
```

Figure 11. An example of spatial filter for three states in the US.

To make the chain generation process automatic, the algorithms introduced in Section 4.4 are adopted. A top-level process (i.e., clip) is created and its built-in sub-processes (i.e., spatial filter and time filter) are assembled in the service chain engine to implement the combination of all involved spatial operations and data for addressing the given spatial question. The embedded spatial operations are encoded as a JSON object. Because in our implementation, OGC standards are heavily applied and followed, the JSON object is then serialized into an XML file containing the full WPS request. This XML file will also be saved in the cyberinfrastructure portal (described in the next section) as the provenance metadata for other researchers to easily reproduce the results by executing the process described in the XML file.

5.4. Integration of a Service Chaining Model into a Cyberinfrastructure Environment

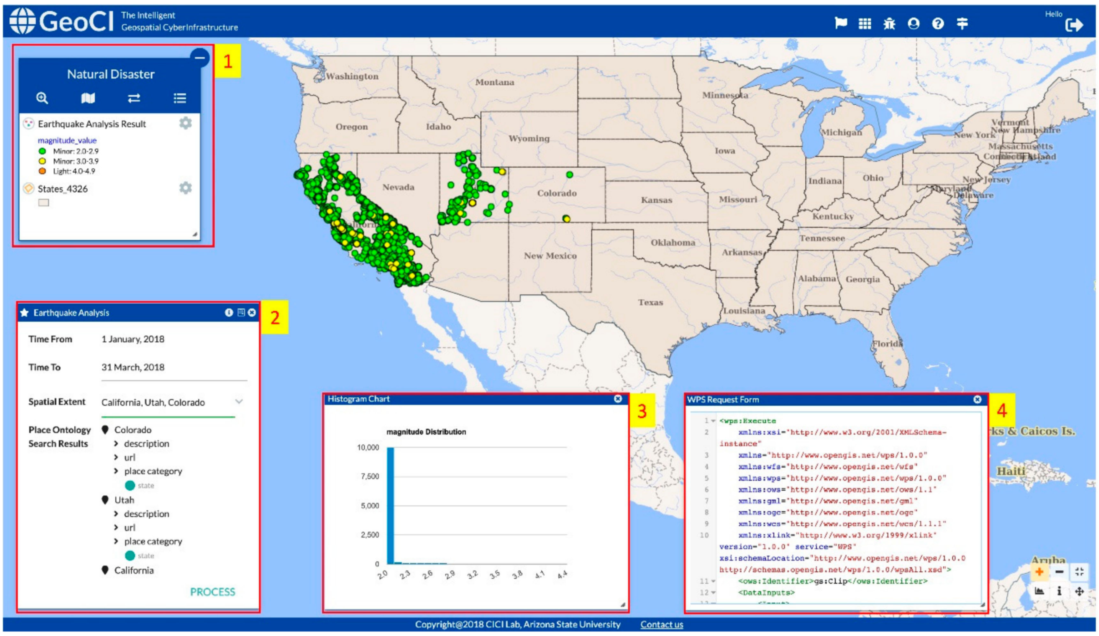
This service chaining model is integrated into the cyberinfrastructure portal (GeoCI; Geospatial CyberInfrastructure: http://cici.lab.asu.edu/gci2/), which supports smart search of geospatial data, seamless integration of disparate online data and analytical resources, and automated question and answering, as well as visual analytics. GeoCI is built upon a service-oriented architecture, and all the data and analytical resources can come from any local or remote services as long as they are compliant with OGC service standards, making this cyberinfrastructure very easy to be extended and scaled. Figure 12 lists different social and environmental use cases supported by the portal. The spatial question can be entered by filling the quadruple <theme><analysis rules><spatial extent><time period> in the UI, which makes a spatial question formalized and understandable for spatial computing.



Earthquake	CALM Sensors	Snow Depth	Vector Data Exploratory Analysis
Projection (Under Development)	WFS Compression Analysis	WFS Compression Test	Vector Tiling Testbed
Advanced Data Resource Search (on going)	Name Entity Extraction	ESDA Exploratory Spatial Data Analysis	Rose Geospatial Distribution Dynamics
Markov Analysis Geospatial Distribution Dynamics	Rank Based Analysis Geospatial Distribution Dynamics	...	...

Figure 12. Use cases enabled in the cyberinfrastructure portal (GeoCI).

Figure 13 shows the results for the earthquake use case given in Section 2. Once the time and place(s) of interest are given (lower left window), the service chaining engine will automatically pull out the corresponding analysis rules from the database and generate an executable metadata file (lower right corner) according to the theme and question of interest. The analytical results (point location of earthquakes) are displayed on the map. A chart showing the statistics of the magnitude of the earthquakes that occurred in the three states of query and the given time period is also generated. For each earthquake event, its attributes can be identified by clicking the point on the map. The map also allows for different operations, such as zoom, pan, or reset. If users are interested in overlaying the current results with another dataset, such as a state boundary, they can conduct a search or select a dataset available from an online web server for integrated analysis.



Earthquake Analysis

Time From 1 January, 2018

Time To 31 March, 2018

Spatial Extent California, Utah, Colorado

Place Ontology Search Results
- Colorado
  - description
  - url
  - place category
    - state
- Utah
  - description
  - url
  - place category
    - state
- California

PROCESS

WPS Request Form

```
<wps:Execute
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns="http://www.opengis.net/wps/1.0.0"
     xmlns:wfs="http://www.opengis.net/wfs"
     xmlns:wps="http://www.opengis.net/wps/1.0.0"
     xmlns:ows="http://www.opengis.net/ows/1.1"
     xmlns:gml="http://www.opengis.net/gml"
     xmlns:ogc="http://www.opengis.net/ogc"
     xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
     xmlns:xlink="http://www.w3.org/1999/xlink"
version="1.0.0" service="WPS"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
     <ows:Identifier>gs:Clip</ows:Identifier>
     <DataInputs>
```

Figure 13. The interactive interface of the cyberinfrastructure portal (GeoCI). The "Natural Disaster" window (label 1) provides the workspace for Earthquake analysis. All data layers and results are displayed in this workspace. The "Earthquake Analysis" window (label 2) provides the interface to allow users to enter interested spatial and temporal constraints of a question on Earthquake in natural language. The window with title "Histogram Chart" (label 3) provides the statistics of the Earthquake data based on some attributes, i.e. magnitude. The executable workflow metadata is demonstrated in the "WPS Request Form" window (label 4).

6. Conclusions

This paper introduces the development of a domain ontology and rule-based service chain model to automatically generate an executable workflow for tackling complex spatial and temporal questions. We built this workflow based on a service-oriented architecture, in which data resources and analytical components are all encapsulated into web services compliant with OGC standards. This design has the advantage of enabling high interoperability among heteronomous resources—the geospatial data of any type, vector, or raster, hosted on any cloud web server, can be easily integrated and utilized to address the research question. The use of standardized web services also fosters the generation of an open knowledge network, in which any application can reuse existing data and analytical resources available on the Web; this will reduce the development cost and accelerate the knowledge creation process. In addition, openly shared data can be leveraged to solve different domain problems, adding value to the data and broadening its impact on society. Moreover, the algorithm introduced to automatically embed data and process for workflow generation is not limited to use for OGC services. We introduced a JSON format to store the workflow metadata, which can be easily converted to different service or workflow standards for integration in a diversity of cyberinfrastructure and desktop applications. The workflow in which the flow of data is encoded provides important provenance information for chains of spatial analysis, ensuring the replicability and validity of one's own and others' research. The novelty of this research also includes its strong focus on spatial sciences and spatial (and temporal) question answering. Many related works [41] in question answering relies on searching a large knowledge graph, in which some already-derived knowledge is encoded. Our work goes beyond question answering in a traditional sense to allow the dynamic creation of knowledge by linking the right data to the right scientific workflow and solve the problem in a collaborative and interoperable manner. The capability of on-demand problem solving makes it stand out from other existing works in the literature.

Although the workflow and service chain can be automatically generated, the analysis rule that each workflow is based on for providing a common data processing ontology continues to be manually created by domain experts. In the future, more data-driven approaches, such as machine learning, will be integrated to automate workflow extraction from the literature. We also plan to integrate a query interface that allows for natural-language (free) -style questions to make the question-answering process even more flexible. Capabilities for advanced visual analytics of the resultant data will be developed to enrich information presentation. A user study will also be conducted to understand the usability of the query formation and user-friendliness of the system. Finally, this workflow generation and execution module will be combined with data quality evaluation module [2] to automatically select the best suitable data for an analysis, and with performance improvement modules [42–44] to enable efficient remote data transfer and visualization in a cyberinfrastructure environment.

Author Contributions: Conceptualization, Wenwen Li; Methodology, Miaomiao Song and Wenwen Li; Software, Miaomiao Song and Yuanyuan Tian; Validation, Wenwen Li, Miaomiao Song and Yuanyuan Tian; Formal Analysis, Wenwen Li, Miaomiao Song and Yuanyuan Tian; Data Curation, Wenwen Li, Miaomiao Song and Yuanyuan Tian; Writing-Original Draft Preparation, Wenwen Li, Miaomiao Song and Yuanyuan Tian; Writing-Review & Editing, Wenwen Li; Visualization, Miaomiao Song and Yuanyuan Tian; Supervision, Wenwen Li.

Funding: This research was funded by National Science Foundation, grant number 1455349, 1937908, 1936677

Acknowledgments: The authors would like to thank Hu Shao for supporting the development of the cyberinfrastructure portal GeoCI.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nosek, B.A.; Alter, G.; Banks, G.C.; Borsboom, D.; Bowman, S.D.; Breckler, S.J.; Buck, S.; Chambers, C.D.; Chin, G.; Christensen, G. Promoting an open research culture. *Science* **2015**, *348*, 1422–1425. [CrossRef]
2. Li, W. Lowering the barriers for accessing distributed geospatial big data to advance spatial data science: The PolarHub solution. *Ann. Am. Assoc. Geogr.* **2018**, *108*, 773–793. [CrossRef]
3. Li, W.; Li, L.; Goodchild, M.; Anselin, L. A geospatial cyberinfrastructure for urban economic analysis and spatial decision-making. *ISPRS Int. J. Geo-Inf.* **2013**, *2*, 413–431. [CrossRef]
4. Anselin, L.; Rey, S.J.; Li, W. Metadata and provenance for spatial analysis: The case of spatial weights. *Int. J. Geogr. Inf. Sci.* **2014**, *28*, 2261–2280. [CrossRef]
5. Foster, I. Service-oriented science. *Science* **2005**, *308*, 814–817. [CrossRef] [PubMed]
6. Demirkan, H.; Delen, D. Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud. *Decis. Support Syst.* **2013**, *55*, 412–421. [CrossRef]
7. Cheatham, M.; Krisnadhi, A.; Amini, R.; Hitzler, P.; Janowicz, K.; Shepherd, A.; Narock, T.; Jones, M.; Ji, P. The GeoLink knowledge graph. *Big Earth Data* **2018**, *2*, 131–143. [CrossRef]
8. Li, W.; Bhatia, V.; Cao, K. Intelligent polar cyberinfrastructure: Enabling semantic search in geospatial metadata catalogue to support polar data discovery. *Earth Sci. Inform.* **2015**, *8*, 111–123. [CrossRef]
9. Shi, X. Where are the spatial relationships in the spatial ontologies? *Proc. Natl. Acad. Sci. USA* **2011**, *108*, E459. [CrossRef] [PubMed]
10. Crawl, D.; Singh, A.; Altintas, I. Kepler webview: A lightweight, portable framework for constructing real-time web interfaces of scientific workflows. *Procedia Comput. Sci.* **2016**, *80*, 673–679. [CrossRef]
11. Honavar, V.G.; Yelick, K.; Nahrstedt, K.; Rushmeier, H.; Rexford, J.; Hill, M.D.; Bradley, E.; Mynatt, E. Advanced Cyberinfrastructure for Science, Engineering, and Public Policy. *arXiv* **2017**, arXiv:1707.00599.
12. Begley, C.G.; Loannidis, J.P.A. Reproducibility in science: Improving the standard for basic and preclinical research. *Circ. Res.* **2015**, *116*, 116–126. [CrossRef] [PubMed]
13. Qi, K.; Gui, Z.; Li, Z.; Guo, W.; Wu, H.; Gong, J. An extension mechanism to verify, constrain and enhance geoprocessing workflows invocation. *Trans. GIS* **2016**, *20*, 240–258. [CrossRef]
14. Zhao, P.; Foerster, T.; Yue, P. The geoprocessing web. *Comput. Geosci.* **2012**, *47*, 3–12. [CrossRef]
15. Di, L. Distributed geospatial information services-architectures, standards, and research issues. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2004**, *35*, 7.

16. Foerster, T.; Stoter, J. Establishing an OGC Web Processing Service for generalization processes. In Proceedings of the ICA Workshop on Generalization and Multiple Representation, Portland, OR, USA, 25 June 2006.
17. Foerster, T.; Lehto, L.; Sarjakoski, T.; Sarjakoski, L.T.; Stoter, J. Map generalization and schema transformation of geospatial data combined in a Web Service context. *Comput. Environ. Urban Syst.* **2010**, *34*, 79–88. [CrossRef]
18. Han, W.; Di, L.; Yu, G.; Shao, Y.; Kang, L. Investigating metrics of geospatial web services: The case of a CEOS federated catalog service for earth observation data. *Comput. Geosci.* **2016**, *92*, 1–8. [CrossRef]
19. Kiehle, C.; Greve, K.; Heier, C. Requirements for next generation spatial data infrastructures-standardized web based geoprocessing and web service orchestration. *Trans. GIS* **2007**, *11*, 819–834. [CrossRef]
20. Stasch, C.; Pross, B.; Gräler, B.; Malewski, C.; Förster, C.; Jirka, S. Coupling sensor observation services and web processing services for online geoprocessing in water dam monitoring. *Int. J. Digit. Earth* **2018**, *11*, 64–78. [CrossRef]
21. Weiser, A.; Zipf, A. Web service orchestration of OGC web services for disaster management. In *Geomatics Solutions for Disaster Management*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 239–254.
22. Meng, X.; Xie, Y.; Bian, F. Distributed Geospatial Analysis through Web Processing Service: A Case Study of Earthquake Disaster Assessment. *JSW* **2010**, *5*, 671–679. [CrossRef]
23. Zhai, X.; Yue, P.; Zhang, M. A sensor web and web service-based approach for active hydrological disaster monitoring. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 171. [CrossRef]
24. Bocher, E.; Petit, G.; Bernard, J.; Palominos, S. A geoprocessing framework to compute urban indicators: The MApUCE tools chain. *Urban Clim.* **2018**, *24*, 153–174. [CrossRef]
25. Meek, S.; Jackson, M.; Leibovici, D.G. A BPMN solution for chaining OGC services to quality assure location-based crowdsourced data. *Comput. Geosci.* **2016**, *87*, 76–83. [CrossRef]
26. Gui, Z.; Wu, H.; Wang, Z. A data dependency relationship directed graph and block structures based abstract geospatial information service chain model. In *Proceedings of the 2008 Fourth International Conference on Networked Computing and Advanced Information Management, Gyeongju, Korea, 2–4 September 2008*; IEEE: Piscataway, NJ, USA, 2008; Volume 2, pp. 21–27.
27. Lemmens, R.; Wytzisk, A.; By, R.; Granell, C.; Gould, M.; Van Oosterom, P. Integrating semantic and syntactic descriptions to chain geographic services. *IEEE Internet Comput.* **2006**, *10*, 42–52. [CrossRef]
28. Yue, P.; Di, L.; Yang, W.; Yu, G.; Zhao, P. Semantics-based automatic composition of geospatial Web service chains. *Comput. Geosci.* **2007**, *33*, 649–665. [CrossRef]
29. Yue, P.; Gong, J.; Di, L. Augmenting geospatial data provenance through metadata tracking in geospatial service chaining. *Comput. Geosci.* **2010**, *36*, 270–281. [CrossRef]
30. Zhao, P.; Di, L.; Yu, G.; Yue, P.; Wei, Y.; Yang, W. Semantic Web-based geospatial knowledge transformation. *Comput. Geosci.* **2009**, *35*, 798–808. [CrossRef]
31. Deng, M.; Di, L. Utilization of Latest Geospatial Web Service Technologies for Remote Sensing Education Through GeoBrain Sysem. In *Proceedings of the 2006 IEEE International Symposium on Geoscience and Remote Sensing, Denver, CO, USA, 31 July–4 August 2006*; IEEE: Piscataway, NJ, USA, 2006; pp. 2013–2016.
32. Li, W.; Yang, C.; Nebert, D.; Raskin, R.; Houser, P.; Wu, H.; Li, Z. Semantic-based web service discovery and chaining for building an Arctic spatial data infrastructure. *Comput. Geosci.* **2011**, *37*, 1752–1762. [CrossRef]
33. Al-Areqi, S.; Lamprecht, A.-L.; Margaria, T. Constraints-driven automatic geospatial service composition: Workflows for the analysis of sea-level rise impacts. In *Proceedings of the International Conference on Computational Science and Its Applications, Beijing, China, 4–7 July 2016*; Springer: Cham, Switzerland, 2016; pp. 134–150.
34. Jelokhani-Niaraki, M.; Sadeghi-Niaraki, A.; Choi, S.-M. Semantic interoperability of GIS and MCDA tools for environmental assessment and decision making. *Environ. Model. Softw.* **2018**, *100*, 104–122. [CrossRef]
35. Scheider, S.; Ballatore, A. Semantic typing of linked geoprocessing workflows. *Int. J. Digit. Earth* **2018**, *11*, 113–138. [CrossRef]
36. Weerawarana, S.; Curbera, F.; Leymann, F.; Storey, T.; Ferguson, D.F. *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging and More*; Prentice Hall PTR: Upper Saddle River, NJ, USA, 2005; ISBN 0131488740.
37. Brauner, J.; Foerster, T.; Schaeffer, B.; Baranski, B. Towards a research agenda for geoprocessing services. In *Proceedings of the 12th AGILE International Conference on Geographic Information Science, Hanover, Germany, 2–5 June 2009*; Leibniz University of Hanover: Hanover, Germany, 2009; Volume 1, pp. 1–12.

38. Yu, G.E.; Zhao, P.; Di, L.; Chen, A.; Deng, M.; Bai, Y. BPELPower—A BPEL execution engine for geospatial web services. *Comput. Geosci.* **2012**, *47*, 87–101. [CrossRef]
39. Zhang, M.; Bu, X.; Yue, P. GeoJModelBuilder: An open source geoprocessing workflow tool. *Open Geospat. Data Softw. Stand.* **2017**, *2*, 8. [CrossRef]
40. Akram, A.; Meredith, D.; Allan, R. Evaluation of BPEL to scientific workflows. In Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid, CCGRID 06, Singapore, 16–19 May 2006; pp. 269–272.
41. Kejriwal, M.; Sequeda, J.; Lopez, V. Knowledge graphs: Construction, querying and management. *Semant. Web J.* **2019**, *10*, 1–2. [CrossRef]
42. Li, W.; Song, M.; Zhou, B.; Cao, K.; Gao, S. Performance improvement techniques for geospatial web services in a cyberinfrastructure environment—A case study with a disaster management portal. *Comput. Environ. Urban Syst.* **2015**, *54*, 314–325. [CrossRef]
43. Shao, H.; Li, W. A comprehensive optimization strategy for real-time spatial feature sharing and visual analytics in cyberinfrastructure. *Int. J. Digit. Earth* **2019**, *12*, 250–269. [CrossRef]
44. Song, M.; Li, W.; Zhou, B.; Lei, T. Spatiotemporal data representation and its effect on the performance of spatial analysis in a cyberinfrastructure environment—A case study with raster zonal analysis. *Comput. Geosci.* **2016**, *87*, 11–21. [CrossRef]